

Docket No.: 003829.P006
Express Mail No.: EM014065324US

UNITED STATES PATENT APPLICATION

FOR

**A METHOD AND APPARATUS TO FACILITATE AUTOMATIC SHARING IN
A CLIENT SERVER ENVIRONMENT**

Inventors:

Tomasz J. Goldman
Claus Tøndering

Prepared By:

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN
12400 Wilshire Blvd., 7th Floor
Los Angeles, California 90025-1026
(310) 207-3800

09881266-061001
003829.P006

A METHOD AND APPARATUS TO FACILITATE AUTOMATIC SHARING IN A CLIENT SERVER ENVIRONMENT

BACKGROUND

Field of the Invention

[0001] The invention relates to shared resources. More specifically, the invention relates to automatic allocation of shares for shared resources in a client-server environment.

Background

[0002] Use of shared resources in a client-server environment is ubiquitous. Typically, when a shared resource, such as storage devices or printers, are attached to the server, the server indicates that those devices are available for sharing by adding the device to a shared resources table. Creation of this table requires a fair amount of overhead and administration. When, for example, a disk is installed, the server goes through the basic input output system ("BIOS") to enable it to recognize the disk. The disk is then known locally and the server goes through its management interface to identify which directories (or the whole disk) are to be shared. Typically, a human administrator is required to identify which disk directories and/or printers to share, assign names to the shared resources and establish access rights. The server then interacts with the clients on the network, using, for example, server message block (SMB) protocol alternatively known as common internet file system (CIFS) protocol to define the shares allocated to various clients. The shared resources table is maintained on the server and when the shared resource device is unplugged

from the server, the corresponding entries from the shared resource table are deleted. Thus, if the device is again attached, the users must reestablish their shares.

BRIEF DESCRIPTION OF THE DRAWINGS

[0003] The invention is illustrated by way of example and not by way of limitation in the figures of the accompanying drawings in which like references indicate similar elements. It should be noted that references to "an" or "one" embodiment in this disclosure are not necessarily to the same embodiment, and such references mean at least one.

[0004] **Figure 1** is a block diagram of the system of one embodiment of the invention.

[0005] **Figure 2A** is a flow diagram of automatic sharing of a storage free device in one embodiment of the invention.

[0006] **Figure 2B** is a flow diagram of automatic sharing of a storage device in one embodiment of the invention.

DETAILED DESCRIPTION

[0007] **Figure 1** is a block diagram of the system of one embodiment of the invention. The server 100 is coupled between shared resources 118, 128 and users of the shared resources such as, local area network ("LAN") nodes 122 and external node 134. LAN nodes 122 are connected to the server by LAN 120. In one embodiment, LAN 120 is an ethernet. External node 134 is coupled over a wide area network such as internet 132 to the server 100. The server 100 provides ports 112 and 114, which may provide various connectivities for the shared resources. In one

embodiment, port 112 is a Universal Serial Bus ("USB") port, such that shared resource 118 is coupled by USB 116 to the server 100. In one embodiment, shared resource 118 is a storage device. In one embodiment, port 114 is an IEEE 1394 port, such that shared resource 128 is coupled by a 1394 connection 126 to the server 100. In one embodiment, resource 128 is a storage free device such as a printer, scanner, camera, etc. Server 100 includes a processor 102 coupled to a memory 104, and a non-volatile memory such as flash 108. Flash 108 contains a descriptor table that identifies storage free devices known to the server. The shared resource table 106 is maintained in the memory 104. In one embodiment, the server is a file server such as the Interjak 200 available from Filanet Corporation of Sunnyvale, California.

[0008] The processor detects connections to its ports 112 and 114. When a shared resource, such as printer 128 is connected, the processor identifies the connection, checks to see if the shared resource is a known resource, (e.g., exists in the descriptor table 110). In one embodiment, the server 100 reads information such as manufacturer, product name and serial number through e.g., a 1394 or USB interface, and compares that information to information in the descriptor table 110. If the shared resource exists in the descriptor table 110, it will automatically be shared under the same name under which it was previously shared. If the printer 128 is unknown, the server automatically creates a unique name under which to share the device based on information such as the manufacturer, product name, or serial number of the device. The name is then added to the descriptor table 110 so the device 128 may be recognized later. While this occurs automatically, in some embodiments, a system administrator may override the automatic name and share

allocations. Moreover, while the above discussion is in the context of a printer, it applies equally to other storage free devices such as scanners, cameras, etc.

[0009] When the processor detects the connection of a storage device, such as disk 118, it automatically queries the device for the presence of a share indicator. If a share indicator is present, the device is a known device and the processor 102 adds its shares as previously defined to the shared resource table 106. In one embodiment, the processor 102 consults the directory 130 on the disk for share allocation information. The share information may be embodied as a share descriptor file which provides a description of which directories on the disk should be shared, under what names those directories should be shared, and any other parameters that govern the sharing. A check sum file may also be present to authenticate the contents of the share file. In one embodiment, the check sum file contains a MD5 check sum of the contents of the share file. If such share information is found, additional shares are allocated according to that information. The allocation is added to the shared resource table 106. If the device is not a known device, the processor assigns the device a unique name to be used when clients want to identify the device subsequently.

[0010] For unknown storage devices, shares may be automatically allocated based on the type of storage device present. For example, in one embodiment, if the device is a CD-ROM (in one embodiment, CD-ROM's are treated as a special case of a storage device), the server automatically creates a read-only browsable share which is the root directory using its volume label as the share name. For a hard disk, the automatically created share may be the root directory that is browsable and read-

write. For portable media, such as Zip drives, the share automatically created may be similar to the CD-ROM case with writability enabled by manual configuration by the administrator. Because the shares are retained in the share file on the media, the generated device and share names survive both accidental and controlled restart of the server 100.

[0011] In one embodiment, while the disk 118 is plugged in, all writes are required to be "write through cache." This reduces the risk of data loss upon unplugging. Also, in one embodiment, the server 100 periodically clears a dirty bit that indicates the "volume mounted" to increase the probability that an unsolicited unplug followed by a hot plug will correctly be interpreted by the server 100 as a hot plug of a correctly dismounted disk. In one embodiment, the clearing occurs at least once per minute. If a resource is unplugged for a period of time, the processor 102 will eventually age out the entry corresponding to that resource from the descriptor table. In one embodiment, entries age out of the descriptor table after seven (7) days. After an entry is aged out, the corresponding device becomes unknown.

[0012] **Figure 2A** is a flow diagram of automatic sharing of a storage free device in one embodiment of the invention. At functional block 202, a shared resource is detected as being connected to the server. At functional block 203, the server obtains device information from the resource. The device information may include product name, manufacturer, serial number, etc. In some embodiments, this information is read directly through the USB or 1994 interface. A determination is made at decision block 204 whether the shared resource is a known device by determining if it is present in the descriptor table. If the shared resource is not in

the descriptor table, the device information is written to the descriptor table. If the shared resource is present or after it is added in the descriptor table, shares are allocated according to the entry in the descriptor table at functional block 210.

[0013] **Figure 2B** is a flow diagram of automatic sharing of a storage device in one embodiment of the invention. At functional block 200, a shared resource is detected as being connected to the server. A determination is made at decision block 206 if the shared resource is writable. For example, is the shared resource a disk, which would be writable, or is this shared resource, e.g., a CD-ROM, which would not be writable. If the device is not writable, a share descriptor file is created in the memory of the server at functional block 208. The share descriptor file, in one embodiment, specifies that the root of the device shall be shared.

[0014] If at decision block 206 the shared resource is writable, the determination is made at decision block 212 whether a share directory is present. If the share directory is present, at decision block 214 a determination is made if a share descriptor file exists. If the share descriptor file exists, a decision is made at decision block 216 whether a check sum file exists. If the check sum file exists, the contents of the check sum file are evaluated at decision block 218 to determine if the check sum is valid. The check sum may be, for example, an MD5 check sum of the contents of the share file. If the check sum is valid at functional block 220, the share descriptor file is read. Then at functional block 222, shares are allocated according to the contents of the share descriptor file. Those shares are then reflected in a share descriptor file in memory and the shared resource table.

[0015] If at any of decision blocks 212 through 218, the directory, file, or check sum are not present or not valid, the server creates a share directory in the root directory of the storage device at functional block 224. In this context, "creates" means to generate if not yet existing and to clear and repopulate if already existing e.g., the case where the directory exists, but the check sum is invalid. At functional block 226, a share descriptor file is created in the share directory. At functional block 228, a check sum file is created in the share directory. The share descriptor file contains the information defining the sharing of the device as discussed above. In one embodiment, the share descriptor file specifies that the root of the device is to be shared. At functional block 230, share information is stored in the share descriptor file. At functional block 232, a check sum is performed on the share file. At functional block 234, the check sum determined is written to the check sum file. Shares of the resource are then allocated according to the contents of the share file at functional block 222.

[0016] In the foregoing specification, the invention has been described with reference to specific embodiments thereof. It will, however, be evident that various modifications and changes can be made thereto without departing from the broader spirit and scope of the invention as set forth in the appended claims. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.